# Quantum Clock Synchronization with a Single Qudit

Armin Tavakoli,[1] Adán Cabello,[2] Marek Żukowski,[3] and Mohamed Bourennane[1]

[1]*Department of Physics, Stockholm University, S-10691 Stockholm, Sweden*
[2]*Departamento de Física Aplicada II, Universidad de Sevilla, E-41012 Sevilla, Spain.*
[3]*Instytut Fizyki Teoretycznej i Astrofizyki, Uniwersytet Gdański, PL-80-952 Gdańsk, Poland.*

Clock synchronization for nonfaulty processes in multiprocess networks is indispensable for a variety of technologies. A reliable system must be able to resynchronize the nonfaulty processes upon some components failing causing the distribution of incorrect or conflicting information in the network. The task of synchronizing such networks is related to detectable Byzantine agreement (DBA), which can classically be solved using recursive algorithms if and only if less than one-third of the processes are faulty. Here we introduce a nonrecursive quantum algorithm that solves the DBA and achieves clock synchronization in the presence of arbitrary many faulty processes by using only a single quantum system.

*Introduction.*—In many multiprocess networks, including data transfer networks, telecommunications networks, and the global positioning system, the individual processes need to have clocks that must be synchronized with one another [1, 2]. To this purpose, individual processes' clocks must periodically be resynchronized. This motivates the need for clock synchronization algorithms which work despite the faulty behavior by some of the processes. Faulty behavior can occur due to a variety of causes, including crashing, transmission failure, and distribution of incorrect or inconsistent information in the network [3]. A clock synchronization algorithm should achieve the following tasks: C1) For any given instant, the time of all nonfaulty processes' clocks must be the same. This is necessary, but not sufficient, since simply stopping all clocks at zero satisfies C1. We therefore need to assume that a process' logical clock also keeps the rate of its corresponding physical clock. In addition, synchronizing may cause further errors, so we require that: C2) There is a small bound on the amount that a process' clock is changed during synchronization [4].

Reliable clock synchronization algorithms can be complicated. To simplify the problem we shall work under the following assumptions [4]: A1) Initially, all clocks are synchronized to the same value. Physical clocks typically do not keep perfect time but drift to respect one another. This motivates the following assumption: A2) All nonfaulty processes' clocks run at one second in clock time per second in real time. A general problem arises from the clocks continuously changing during the synchronization procedure. Unless the synchronization algorithm is very fast, this will cause problems. This motivates our last assumption: A3) A nonfaulty process can read the time difference between the clock of another process and its own.

A method to achieve synchronization is to use interactive consistency algorithms (ICAs) in which all nonfaulty processes reach a mutual agreement about all the clocks [4]. A ICA should satisfy that, for every process $p$: (1) Any two nonfaulty processes obtain the same value of process $p$'s clock, even if $p$ is faulty. (2) If $p$ is nonfaulty, then every nonfaulty process obtains the value of $p$'s clock.quit

The conditions for ICAs make them suitable for the task of fault tolerant synchronization. For most applications it is sufficient to consider a scenario called detectable Byzantine agreement (DBA) or detectable broadcast [5, 6]. In this case, it is required that: (i) either all nonfaulty processes obtain the same value or all abort, and (ii) if process $p$ is nonfaulty, then either every nonfaulty process obtains the same value or aborts. By "abort" we mean treating the value as undefined and exiting the protocol.

Classical ICAs can only achieve fault tolerant synchronization through DBA if less than one-third of the processes are faulty [4] and agreement is achieved by majority voting using a recursive algorithm, called $OM(n)$, where $n$ is the number of faulty processes. The $OM(n)$ algorithm works as follows. We label the processes as $P_k$, with $k = 1, 2, \ldots, m$. If $n = 0$, then $P_1$ distributes its value to every other process. Every process uses the value received from $P_1$ and, in case no value is obtained, uses 0. If $n > 0$, then $P_1$ distributes its value to every other process. For $k = 2, \ldots, m$, let $x_k$ denote the value obtained by $P_k$ from $P_1$. If $P_k$ receives no message, then let $x_k = 0$. $P_k$ acts as $P_1$ in algorithm $OM(n-1)$ by distributing $x_k$ to the remaining $m-2$ processes. For every $k$ and $\forall j \neq k$, let $x_j$ be the value received by $P_k$ from $P_j$ using $OM(n-1)$, and in case no value was received $x_j = 0$. $P_k$ decides on the value obtained from the median of $(x_1, \ldots, x_m)$. Thus, $OM(n)$ requires $O(m^{n+1})$ transmitted messages to solve the task.

The DBA is an example of a communication task for which quantum resources can provide a solution, while classical tools cannot. Nevertheless, the sepcial case of DBA in a three process network where one is faulty, has been solved using quantum methods based on three-qutrit singlet states [5, 7], four-qubit entangled states [8, 9], and three [6] or two [10] pairwise quantum key distribution (QKD) channels.

Interestingly, later works have shown that there are quantum solutions for certain communication complexity problems and secret sharing tasks which do not require entanglement, but, instead, sequential communication of a single quantum system [11, 12]. These protocols have been shown to be much more resistant to noise and imperfections, and significantly more scalable than protocols based on entanglement.

In this paper, we introduce a quantum ICA that solves the DBA and achieves clock synchronization in the presence of an arbitrary number of faulty processes, with only one single round of message passing per process independently of the number of faulty processes, utilizing only a single quantum system.

In order to solve the DBA problem, the $m$ processes need to share data in the form of lists $l_k$, of numbers subject to specific correlations, and the distribution must be such that the list $l_k$ held by process $P_k$ is known only by $P_k$. Quantum mechanics provides methods to generate and securely distribute such data, here we shall seek for one which is simple, efficient, and easily extendible to an arbitrary number of processes. We assume that all processes can communicate with one another with oral messages by pairwise authenticated error-free classical channels and pairwise authenticated quantum channels.

*Correlated lists and their use.*—The initial stage of the quantum protocol is to distribute lists $l_k$, for $k = 1, \dots, m$, each of them available only to process $P_k$. All lists have to be of the same length $L$ and are required to satisfy the property that if $N = 0$ (or 1) is at position $j$ in $l_1$, then 0 (respectively, 1) is at position $j$ in lists $l_k$ for $k = 2, \dots, m$ (i.e., they are perfectly correlated). However, if $N \in \{2, \dots, m-1\}$ is at position $j$ in $l_1$, then the sum of numbers at positions $j$ in lists $l_k$ for $k = 2, \dots, m$ equals $m - N$, and all elements in these lists are either 0 or 1. Given an $N$, all the possible combinations of binary numbers satisfying the condition are uniformly probable.

Note that, on one hand, $P_1$ has information about at which positions the lists of all other processes the values are perfectly correlated, and at which positions they are random bits, with the property that their sum is anti-correlated with the value, $N \geq 1$, in $l_k$. On the other hand, the holder of one the lists $l_k$, with $k = 2, \dots, m$, has no information whatsoever on whether the lists are correlated at a given position or not.

Once the processes have these lists, they can use them to achieve mutual agreement and solve the DBA by applying the algorithmic part of the protocol, which we shall call $QB(n, m)$. The special case, $QB(1, 3)$, reproduces the protocol in [9].

(1) $P_1$ sends bit-valued messages to all processes. The message sent to process $P_k$ will be denoted by $m_{1,k}$. Together with each message, $P_1$ sends a list $l_{1,k}$ of all of the positions in $l_1$ in which the value $m_{1,k}$ appears. If $P_1$ is nonfaulty all lists and messages are identical. The full information which $P_k$ receives from $P_1$ will be denoted by $\{m_{1,k}, l_{1,k}\}$.

(2) The receiving processes $P_k$ analyze (singlehandedly) the obtained lists and messages. If the analysis of $P_k$ shows that $l_{1,k}$ is of appropriate length (i.e., about $L/m$) and $\{m_{1,k}, l_{1,k}\}$ is consistent with $l_k$ at all positions, then if $P_k$ is nonfaulty, it conveys $\{m_{1,k}, l_{1,k}\}$ to all other processes $P_{k\neq 1}$. A faulty process sends a flipped bit value of the message with a whatever list it chooses. The full information which $P_j$ receives from $P_k$ will be denoted by $\{m_{k,j}, l_{k,j}\}$.

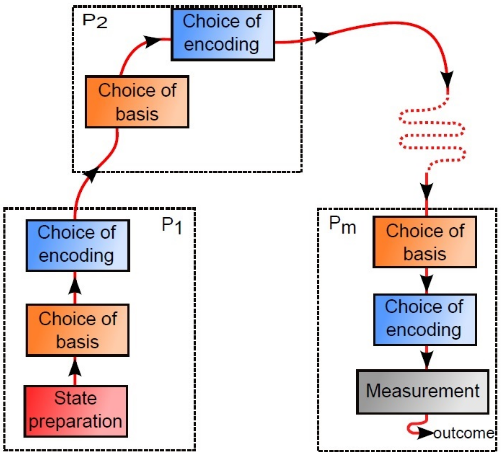

FIG. 1. Scheme of the quantum protocol for the distribution of the correlated lists. $P_1$ prepares a uniform $d$-level superposition state, makes a choice of basis and encoding, and forwards the qudit to $P_2$ which applies a choice a basis and encoding and forwards the qudit to $P_3$. Processes $P_3, ..., P_m$ act in analogy with $P_2$. Finally $P_m$ projects the state onto the initial state prepared by $P_1$ and if the outcome is 1 the round is treated as valid.

A nonfaulty $P_k$ will also decide on the final bit value it adopts $V_k$. This is $m_{1,k}$, unless messages from the other processes force it to decide that $P_1$ is faulty. However, if $\{m_{1,k}, l_{1,k}\}$ is not consistent with $l_k$, then $P_k$ immediately ascertains that $P_1$ is faulty and relays to other processes neither 0 nor 1 but $\perp$, meaning "I have received inconsistent data."

(3) Once all messages have been exchanged between $P_2, \dots, P_m$, each process considers the obtained data and acts according to the instructions in Table I. The overall aim is, if $P_1$ is nonfaulty, to have the same value of $V_k$ for all nonfaulty processes, or all of them aborting.

*Quantum protocol for distributing lists $l_k$.* All processes are equipped with devices which can unitarily transform qudits. In addition, $P_1$ has a source of *single qudits of dimension* $m$ and the last process, $P_m$, has *additionally* a measurement device. The protocol runs as follows (for an illustration, see Fig. 1):

(I) $P_1$ prepares the state

$$|\psi_0\rangle = \frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} |j\rangle. \tag{1}$$

TABLE I. Once $P_k$ receives all messages and lists from all other processes, it will study the obtained lists and messages and compare to its own list $l_k$. Depending on the consistency between obtained and private data $P_k$ will act according to table below. Notation $\{m_{j,k}, l_{j,k}\} \cong l_k$ means that $m_{j,k}$ and $l_{j,k}$ are found to be consistent with $l_k$ whereas $\ncong$ means "inconsistent with." The symbol $\perp$ means "I have received inconsistent data." By $\mathbb{M}_k$ we denote some non-empty subset of $\{1, \ldots, m\} \setminus \{k\}$.

| | local analysis of all data received by $P_k$ | decision of $P_k$ on the value $V_k$ |
|---|---|---|
| (iia) | $\forall j \in \mathbb{N}_m \setminus \{k\}$, $\{m_{j,k}, l_{j,k}\} \cong l_k$ and all messages are equal | $V_k = m_{1,k}$, no faulty process |
| (iib) | $\forall j \in \mathbb{N}_m \setminus \{k\}$, $\{m_{j,k}, l_{j,k}\} \cong l_k$ and *not* all messages are equal | as $P_1$ is faulty, $V_k = abort$ |
| (iic) | $\forall j \in \mathbb{M}_k$, $\{m_{j,k}, l_{j,k}\} \ncong l_k$ and $\forall j \notin \mathbb{M}_k$, $\{m_{j,k}, l_{j,k}\} \cong l_k$ | $V_k = m_{j,k}$, for $j \notin \mathbb{M}_k$, as the other $P_j$'s are faulty |
| (iid) | $\forall j \in \mathbb{M}_k$, $\{m_{j,k}, l_{j,k}\} \cong l_k$ and $\perp$ $\forall j \notin \mathbb{M}_k$ | $V_k = m_{j,k}$, although $P_1$ could be faulty |
| (iie) | $\forall j \in \mathbb{M}_k$, $\{m_{j,k}, l_{j,k}\} \cong l_k$, but with unequal messages, and $\perp$ from $\forall j \notin \mathbb{M}_k$ | $V_k = abort$, at least $P_1$ is faulty |

(II) $P_1$ randomly chooses the "encoding basis" from $m$ different options $U_0, ..., U_{m-1}$ and labels the choice $c_1$. Having chosen the $c_1$'st encoding basis, process $P_1$ applies the following unitary transformation to the qudit:

$$U_{c_1} = |0\rangle\langle 0| + \sum_{k=1}^{m-1} \omega^{c_1} |k\rangle\langle k|, \tag{2}$$

where $\omega = e^{i\frac{2\pi}{m}}$. From the interferometric point of view, applying $U_{c_1}$ introduces a phase-shift of $-2\pi c_1/m$ in the first beam.

(III) After that, $P_1$ randomly chooses a value $N_1$ in the set $\{0, 1, \ldots, m-1\}$ and encodes $N_1$, by applying the following unitary transformation:

$$U(N_1) = \sum_{j=0}^{m-1} \omega^{jN_1} |j\rangle\langle j|. \tag{3}$$

Afterwards, the qudit is sent to $P_2$.

(IV) $P_2$, in the same manner as $P_1$, choses a $c_2 \in \{0, ..., m-1\}$ and applies a the unitary $U_{c_2}$ corresponding to choice of encoding basis.

(V) Next, $P_2$ randomly chooses a value $N_2$ in the set $\{0, 1\}$. If $N_2 = 0$, no action is taken, i.e., $P_2$ applies the transformation $U(N_2 = 0) = \mathbf{1}$. If $N_2 = 1$, then $P_2$ applies $U(N_2 = 1)$ and then sends the qudit to $P_3$.

(VI) $P_3, \ldots, P_m$ consecutively repeat the same procedure as $P_2$ with independent choices of basis and encoding their respective random values $N_3, \ldots, N_m$.

(VII) In addition, $P_m$ measures the qudit using a device which distinguishes the state $|\psi_0\rangle$ from any set states orthogonal to it.

(VIII) If $P_m$ obtains $|\psi_0\rangle$, then the processes consecutively reveal their encoding bases (but not their values $N_k$) in reverse order: First $P_m$ and last $P_1$. If it turns out that the sum of the basis choices modulo $m$ equals zero, then the run is treated as a valid distribution of the numbers $N_k$ at the same position in the private lists $l_k$.

The protocol distributes the numbers in the required way because all the unitary operators are diagonal and, therefore, commute. Additionally, if $\sum_{k=1}^{m} c_k = 0$ mod $m$ then

$$\prod_{k=1}^{m} U_{c_k} = \mathbf{1}, \tag{4}$$

and, if $\sum_{k=1}^{m} N_k = 0$, modulo $m$, then

$$\prod_{k=1}^{m} U(N_k) = \mathbf{1}. \tag{5}$$

Whenever this condition is not satisfied, the final state of the system is orthogonal to $|\psi_0\rangle$ and will therefore never be an outcome of $P_m$'s measurement.

*Clock synchronization.*—Fault tolerant clock synchronization is one possible adaption of our method to achieve DBA. However, in this case, a problem arises from clocks ticking during the synchronization procedure. This is solved by exploiting assumption A3: Instead of sending a number, the processes send their clock differences to each other. In the classical case, we achieve clock synchronization by running the algorithm $OM(1)$ $m$ times, sending clock differences instead of the binary values, and analogously for $OM(n)$ [4]. In analogy with the classical case, the processes send clock differences also in the quantum case, exploiting the fact that the clock differences can be decomposed into binary strings up to arbitrary accuracy agreed upon in advance. We run $QB(n, m)$ $m$ times in such a way that for each run a new processes takes the roll of $P_1$ in $QB(n, m)$. More explicitly, $P_y$ reads the clock difference $\Delta_{xy}$ between its own clock and the clock of $P_x$. If $P_y$ is nonfaulty it will relay $\Delta_{xy}$ to $P_z$ but if $P_y$ is a faulty process, it can arbitrarily change $\Delta_{xy}$ before sending it. If $P_y$ relays the value obtained from $P_x$ to $P_z$, then $P_z$ knows the time difference between $P_x$ and $P_y$. Also, since $QB(n, m)$ is ran $m$ times, $P_z$ will also obtain $\Delta_{yz}$ from $P_y$ and thus $P_z$ knows that $P_y$ is claiming that the time difference between $P_x$ and $P_z$ is $\Delta_{xy} + \Delta_{yz}$, which can then be compared to $\Delta_{xz}$ obtained directly from $P_x$.

*Comparison with the other solutions.*—The correlated lists needed for achieving DBA can be distributed by other means than with the single-qudit protocol. Successful distribution can be achieved by the process $P_m$

sharing a QKD channel with every other process. $P_m$ uses a QKD protocol, e.g., BB84 [15] to distribute numbers such that (1) $P_m$ and $P_1$ share a string $K_{1,m} = k^1_{1,m} \dots k^L_{1,m}$, where $k^j_{1,m} \in \{0, \dots, m-1\}$. (2) For every $l = 2, \dots, m-1$, $P_m$ and $P_l$ share a string $K_{l,m} = k^1_{l,m} \cdots k^L_{l,m}$ such that $k^j_{l,m} \in \{0,1\}$. (3) For a given $j$, the lists satisfy $(\sum_{l=1}^{m} k^j_{l,m})_{\text{mod m}} = 0$. (4) None of $P_2, \dots, P_{m-1}$ have any information about a particular list element of any other process. (5) Whenever $P_1$ receives an element $k^j_{1,m} \geq 2$, $P_1$ has no information on the bit value of $k^j_{l,m}$ for $l = 2, \dots, m$, and whenever $P_1$ receives $k^j_{1,m} = p \in \{0,1\}$, $P_1$ knows that $k^j_{l,m} = p$ for all $l = 2, \dots, m$. All QKD channels except that shared between $P_1$ and $P_m$ transmit bit values. In order to transmit elements of $\{0, \dots, m-1\}$ to $P_1$, the numbers must be encoded into $\lceil \log_2(m) \rceil$ qubits. One additional requirement that has to be made for solving the DBA using the QKD distributed lists is that $P_m$ is not required to convey any lists. This is necessary since $P_m$ has full knowledge about the lists of all other processes and therefore easily could cheat. Instead, $P_m$ may announce the message it received from $P_1$, and if any inconsistency is noted by $P_2, \dots, P_{m-1}$, then $P_m$ will change its final value if the other processes convince $P_m$ of them being nonfaulty.

There is also a number of proposed solutions to the DBA considering three processes where one is faulty. The first one, proposed in Ref. [5], relies on the three qutrit entangled Aharonov state. The goal is to distribute lists given by all permutations of the elements of the set $\{0, 1, 2\}$, i.e., (0–1–2, 0–2–1, 1–0–2, 1–2–0, 2–0–1, and 2–1–0). Generalization to $m$ parties along the lines of [5] would require the usage of multipartite $m$-level entanglement, provided by the state

$$|\kappa_m\rangle = \frac{1}{\sqrt{m!}} \sum_{\bar{i}=\sigma(S_m)} (-1)^{N(\sigma(S_m))} |i_1, \dots, i_m\rangle, \qquad (6)$$

where $\bar{i} = \{i_1, \dots, i_n\}$, $S_m = \{0, \dots, m-1\}$ and $N(\sigma(S_m))$ is the parity of the permutation of $S_m$. Already for the simplest case of $m = 3$, this approach requires the preparation of a very complex state which, to our knowledge, has not yet experimentally realized. However, for the three process case, it has been pointed out in [10] that the distribution of the lists can be realized without the state (6), by utilizing two separated QKD channels. With small modification for the $m$ process setting, distribution of the lists is achieved with $m-1$ QKD channels. However, to encode the entire space provided by $S_m$, the QKD requires $\lceil \log_2(m) \rceil$ qubits. If the efficiency of a detector $\eta$ is not perfect and the QKD is performed with single qubits using von Neuman measurements, successful distribution occurs only with probability $\eta^{(m-1)\lceil \log_2(m) \rceil}$. Typically, the classical part of the protocol in [5] and its possible generalizations scale rapidly with the number of processes. It is required that $m!$ different types of lists are distributed. However, a solution to the three party DBA exploiting four-qubit entanglement provides a simpler classical part of the protocol: the number of different lists is lowered from six to four [9].

The general $m$ process protocol presented in this paper generalizes the protocol in [9] and requires $2^{m-1}$ different types of lists. As emphasized earlier, the distribution of the required lists can be achieved both with single-qudit and with $m-1$ QKD channels. Using QKD channels, only one channel needs to transmit all elements in $S_m$ while the remaining $m-2$ channels only transmit bit values. In the presence of nonperfect detectors, successful distribution occurs with probability $\eta^{m-2+\lceil \log_2(m) \rceil}$. However, in the single-qudit approach only one single detection is needed and, therefore, successful distribution of the lists occur with probability $\eta$ independently of $m$. The single-qudit protocol is highly scalable, both in terms of success probability with inefficient detectors and requirements on the classical lists.

*Conclusions.*—We have presented a single-qudit protocol which provides an efficient solution to an important multiparty communication problem: It solves DBA and achieves clock synchronization in the presence of arbitrary many faulty clocks. In principle, our quantum algorithm is not limited to the case of clock synchronization, it can with small modification be used for other tasks requiring oral message interactive consistency. Interestingly, our algorithm works by transmitting a single qudit among the parties rather than by distributing a quantum entangled state among them. This makes the protocol much more practical, as single qudits can be experimentally realized easily in many ways. For example, using unbiased multiport beamsplitters [13] or time-bin [14]. Compared to schemes based on several QKD channels, the single-qubit protocol is more scalable and robust against detection inefficiencies. This results shows that single-qudit quantum information protocols are interesting beyond QKD [16–18] and random number generation [19, 20], and should stimulate experimental implementations and further research in quantum information protocols.

**Acknowledgements:** This project was supported by the Swedish Research Council, ADOPT, the Project No. FIS2011-29400 (MINECO, Spain) with FEDER funds, the FQXi large grant project “The Nature of Information in Sequential Quantum Measurements,” MNiSW Grant No. IdP2011 000361 (Ideas Plus) and Foundation for Polish Science TEAM project co-financed by the EU European Regional Development Fund. The work is an extension of the first sketch presented in [21].

[1] Simons, B. Welch,J. L. & Lynch N. An overview of clock synchronization. *Fault-Tolerant Distributed Computing, Lecture Notes in Computer Science* **448**, 84-? (Spinger, New York, 1990).

[2] Lewandowski, W., Azoubib, J. & Klepczynski W. J. GPS: Primary tool for time transfer. *Proc. IEEE* **87**, 163-172 (1999).

[3] Lamport, L. & Melliar-Smith, M. Byzantine Clock Synchronization. *Proc. of the 3rd Ann. ACM Symposium on Principles of Distributed Computing (PODC 1984)*, 68-74 (ACM Press, New York, 1984).

[4] Lamport, L. & Melliar-Smith, M. Synchronizing clocks in the presence of faults. *J. ACM* **32**, 52-78 (1985).

[5] Fitzi, M., Gisin, N. & Maurer U. A quantum solution to the Byzantine agreement problem. *Phys. Rev. Lett.* **87**, 217901 (2001).

[6] Fitzi, M., Gottesman, D., Hirt, M., Holenstein, T. & Smith, A. Detectable Byzantine agreement secure against faulty majorities. *21th ACM Symposium on Principles of Distributed Computing (PODC 2002)*, 118-126 (ACM Press, New York, 2002).

[7] Cabello, A. $N$-particle $N$-level singlet states: Some properties and applications. *Phys. Rev. Lett.* **89**, 100402 (2002).

[8] Cabello, A. Solving the liar detection problem using the four-qubit singlet state. *Phys. Rev. A* **68**, 012304 (2003).

[9] Gaertner, S., Bourennane, M., Kurtsiefer, C., Cabello, A. & Weinfurter, H. Experimental demonstration of a quantum protocol for Byzantine Agreement and Liar Detection. *Phys. Rev. Lett.* **100**, 070504 (2008).

[10] Iblisdir S. & Gisin, N. Byzantine agreement with two quantum key distribution setups. *Phys. Rev. A* **70**, 034306 (2005).

[11] Schmid, C., Trojek, P., Bourennane, M., Kurtsiefer, C., Żukowski, M. & Weinfurter, H. Experimental single qubit quantum secret sharing. *Phys. Rev. Lett.* **95**, 230505 (2005).

[12] Trojek, P., Schmid, C., Bourennane, M., Brukner, Č., Żukowski, M. & Weinfurter, H. Experimental quantum communication complexity. *Phys. Rev. A* **72**, 050305 (2005).

[13] Żukowski, M., Zeilinger, A. & Horne, M. A. Realizable higher-dimensional two-particle entanglements via multiport beam splitters. *Phys. Rev. A* **55**, 2564 (1997).

[14] Marcikic, I., de Riedmatten, H., Tittel, W., Scarani, V., Zbinden, H. & Gisin, N. Time-bin entangled qubits for quantum communication created by femtosecond pulses. *Phys. Rev. A* **66**, 062308 (2002).

[15] Bennett C. H. & Brassard, G. *Proceedings of the International Conference on Computers, Systems and Signal Processing, Bangalore, 1984*, 175-179, (1984).

[16] Cerf, N. J., Bourennane, M., Karlsson, A. & Gisin, N. Security of quantum key distribution using $d$-level systems. *Phys. Rev. Lett.* **88**, 127902 (2002).

[17] Cabello, A., D'Ambrosio, V., Nagali, E. & Sciarrino, F. Hybrid ququart-encoded quantum cryptography protected by Kochen-Specker contextuality. *Phys. Rev. A* **84**, 030302(R) (2011).

[18] Svozil, K. *Physics and Computation 2010, edited by H.Guerra (University of Azores, Portugal, Ponta Delgada, 2010)*, 235-249 (2010).

[19] Svozil, K. Three criteria for quantum random-number generators based on beam splitters. *Phys. Rev. A* **79**, 054306 (2009).

[20] Um, M. *et al.* Experimental certification of random numbers via quantum contextuality. *Sci. Rep.* **3**, 1627 (2013).

[21] Bourennane, M., Cabello, A. & Żukowski, M., arXiv:1001.1947, (2010).